Ingest

Guide Navigation

1. The Basics
2. ECHO spatial representations
3. Metadata model
4. **Ingest**
5. Data management
6. Fulfilling orders
7. Getting started
8. Acronyms

Ingest

Ingest Overview

Ingest refers to the process of inserting, deleting, updating, or validating metadata in the ECHO database and affects only that Data Partner's specific metadata. Data Partners may generate a full metadata record that will be processed as an insert or full replacement. Partners may generate metadata to update or delete specific fields within a metadata record or the entire metadata record from ECHO. Partners may also generate reconciliation metadata to perform a full metadata verification or inventory verification.

Note that deleting a collection will cascade to delete all associated granules. However, a cascaded or explicit granule deletion will not cascade to delete associated browse records.
The ECHO Ingest process directly accepts and processes metadata input files conforming to the ECHO 10 format and it is the responsibility of the Data Partner to ensure their metadata conforms to this specification. Unless supported by previous versions of Ingest, metadata input files conforming to other formats require that you perform a conversion before submitting the files to the ECHO ingest process. Contact the ECHO Operations team for more information.

The main tasks for the metadata ingest process are loading metadata and updating the ECHO database. Although there are slight variations from one Data Partner to another in the transmission process and interaction with ECHO Operations, the process includes the following steps. For full details, please refer to the additional information found in this section.

1. Data Partners send metadata xml files and browse image files to a configured FTP location on the ECHO system.
2. ECHO Ingest regularly monitors each Data Partner's ingest ftp directory for new or updated metadata based upon an Operationally configured interval. When Ingest detects metadata which has been transmitted and quiesced for a configured amount of time, Ingest will create a new Ingest job and remove the files from the ftp directory.
    a. If the single-file delivery mechanism is used, received files will be aggregated into a single job.
    b. If the package delivery mechanism is used, each package will become a unique Ingest job.
3. ECHO ingest will process jobs based upon the order received for jobs created from the single-file delivery mechanism or packages which do not specify a sequence number. If packages specify a sequence, Ingest will process jobs in the specified order.
4. ECHO Ingest will validate the input file against the ECHO 10 Format and business rules. If this validation fails, the ingest process will reject the input file if it cannot be parsed or reject only the invalid metadata items found in an input file. Whenever the ingest process rejects an input file or metadata item, it will record an error and include the error in the Ingest Detail Report.
5. Once metadata items in an input file have been successfully validated, the ECHO ingest process will load the metadata into the ECHO database, making the metadata available for public search. To ensure proper metadata processing, only one ingest job may be in this 'loading' state at a time. The ingest process will send an end of job email to a preconfigured provider contact email address. This email will contain information regarding successful and unsuccessful metadata actions. An xml Ingest Summary Report file is also placed in an ftp accessible location for the Data Partner to retrieve.

The following diagram shows the full flow of ingested data beginning at Data Partner ftp submission through report creation and email.

Ingest Processing States

During the Ingest process, an ingest job will pass through numerous internal states. These states are described below. Data Partners will not be notified as a job passes through each state. However, through the ECHO Ingest Accounting Tool, a Data Partner may view the current status of a job.

1. RESOURCE_WAITING - Indicates the job is currently waiting for further resources to arrive from the provider. Resources currently include browse image files.
2. SEQUENCE_WAITING - Indicates the job is waiting on a sequence number earlier in the list to arrive and be processed.
3. INPUT_ADAPTING - Indicates the job is having some of its files adapted from a provider specific format.
4. SORTING - Indicates the job is having its files sorted by item type and action.
5. LOAD_WAITING - Indicates the job is waiting for another job from the same provider to finish loading.

6. LOADING - Indicates the job's metadata files are currently being validated and loaded into the database.
7. REPORTING - Indicates a report is being generated for the job.
8. CLEANING_UP - Indicates that cleanup is occurring on the job.
9. COMPLETED - Indicates the job has been completed.

IngestProcessingOrder

In order to ensure metadata records are processed correctly and consistently, ECHO Ingest will sort received metadata files and process actions in the following order:

1. Browse inserts/replacements
2. Collection inserts/replacements
3. Collection partial deletes
4. Collection partial updates
5. Collection deletes
6. Granule inserts/replacements
7. Granule partial deletes
8. Granule partial updates
9. Granule deletes
10. Browse deletes
11. Collection metadata verifications
12. Granule metadata verifications
13. Collection inventory verifications
14. Granule inventory verifications
15. Browse inventory verifications

Delivering Data to ECHO for Ingest

Data Partner FTP Locations

The ECHO Operations team configures an ftp directory for each Data Partner utilizing a specific username and password. Data Partners have permissions to write data to this input directory, but cannot remove files. If file removal is needed, the ECHO Operations team (echo@echo.nasa.gov) should be contacted. Note that ECHO Ingest will support filenames up to 255 characters long.

> Due to security constraints, ftp access to ECHO resources is restricted based upon the originating IP address. Initial configuration or subsequent changes in test or operational systems will require Data Partners to coordinate with ECHO Operations to ensure the proper provisions are made.

Package File Delivery

Package delivery is the preferred Data Partner delivery mechanism. Data Partners deliver a package that is a ZIP archive of metadata files. The archive contains metadata files of any action and type as well as a manifest file that provides a list of the files in the package and an optional sequence number. Packages have the benefit of compression, which greatly reduces the amount of data delivered. They also support optional sequencing that allows you to indicate the order that the packages should be executed in—which is important in the event that packages arrive out of order and have dependencies on other packages.

ECHO Ingest guarantees that a single package will map to a single job with all the files in the package included in the job. Browse binary files are still delivered externally to the archive due to the size and number of the files involved.
If package sequencing is being used by a provider, packages with a sequence number of '-1' will be processed on a first come first serve basis, outside of the regularly sequenced packages.

CodeListing38.ManifestFileExample

Code Listing 38. Manifest File Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Manifest packageName="1101" sequenceNumber="1">
    <Files>
        <File name="CollectionInsertFile.XML"/>
        <File name="GranuleInsertFile.XML"/>
    </Files>
</Manifest>
```

The manifest file for package delivery must be named "manifest.xml" - all lowercase letters.

For more information, refer to the package manifest schema and documentation located on the ECHO website (http://www.echo.nasa.gov/ingest/schemas/operations/docs/index.html)

Single File Delivery

Single file delivery is a legacy delivery mechanism. This mechanism allows you to deliver uncompressed metadata files one at a time. ECHO Ingest will monitor the delivered files for completion based on arrival date. Once a file is determined to be complete, a job will be constructed and the file added.

ECHO Ingest will attempt to put as many files into a single job as possible, but cannot guarantee that all of the delivered files will be in a single job because some files may not be considered complete when the input scan is performed. It is therefore important for you to deliver the files serially in the order that they desire execution. Once a job is created, ECHO Ingest will generate an internal sequence number to ensure that individual jobs are executed in the order they were created.

Metadata Generation Tips

CreatingECHO-CompatibleXMLFiles

The ECHO Ingest process directly accepts and processes metadata input files conforming to the ECHO 10 schema. Metadata input files conforming to other DTDs require that you perform an XML-to-XML conversion before submitting the files to the ECHO ingest process.

When the ECHO ingest process detects potential input files in your FTP input directory, it will then examine and validate each file, checking to see if the first line of text contains the <!xml....> declaration. If this line is not present, Ingest will reject the input file as invalid.

Dates and Times

Providers utilizing the ECHO 10.0 XML schema format must conform to the XML Schema Specification. Unless otherwise specified in the provided date/time value, ECHO will process all date and time fields in the UTC timezone.

Unique Values

ECHO requires uniqueness in specific metadata fields in order to uniquely identify metadata items within a Data Partner's holdings. These fields are validated for uniqueness during ingest and used during updates, replacements, and deletions in order to uniquely a target record. The fields which are validated for uniqueness are found below.

- **(Collection) DatasetID** - This field used as the primary identifier of a collection.
- **(Collection) ShortNameandVersionID** - The combination of these two fields must be unique.
- **(Collection) LongName** - This field must be unique.
- **(Granule) GranuleUR** - This field is used as the primary identifier of a granule.
- **(Browse) ProviderBrowseId** - This field is used as the primary identifier of a browse image.

Metadata Inheritance

Specific metadata fields have been designated to require an enforced 'inheritance' validation between collections and associated granules.

This means that values for these fields which are specified at the collection level include a superset of all values which may be specified at the granule level. For example, if a collection specifies Campaigns A, B, and C, all associated granules may have any subset of these campaigns. However, if a granule specifies a Campaign D, it will be rejected. Collection and granule updates will always perform this validation check to ensure the inheritance is valid. Affected fields include the following:

- Campaign
- Platform
- Instrument
- Sensor
- Additional Attributes

New Items vs Replace Items

ECHO ingest requires a complete set of collection, granule or browse metadata for both inserts and replacements. When processing collection, granule, or browse metadata, the ECHO system first checks to see if the item already exists. Item existence is based upon the primary unique identifier for each item type, as described in section 5.3.2. If the item already exists in the system, ECHO will replace the metadata associated with the item with the new metadata. Otherwise, ECHO will consider the metadata item to be an insert and load the new item into the ECHO database. The ECHO system applies the same principles and validation when dealing with the insertion or replace of the collection, granule, or browse item when processed against XML metadata input.

When replacing collection, granule, or a browse item, only the version with the most recent last update time will be stored in the ECHO database. ECHO will reject any collection, granule, or browse item received that has a last update date pre-dating the records already in the database. If there are duplicated items in one input file the same rule applies, the most recent item will be ingested into ECHO and earlier versions of the item may be reported as rejected, depending on the order of items processed within the file. If the last update time of the item is the same as the one currently in the database, the replacement will be applied. Due to backward compatibility with previous versions of Ingest, the last update time threshold is +/- a half a second.

Code Listing 39. Metadata Update Example

```
<GranulePartialAdd>
  <Targets>
    <Target>
      <GranuleUR>GR:115628</GranuleUR>
    </Target>
  </Targets>
  <Fields>
    <Field>
      <BrowseImage>BR:036015124</BrowseImage>
    </Field>
  </Fields>
</GranulePartialAdd>
```

Non-Replaceable Fields

The following information describes situations where ECHO will reject a processed replacement upon dependent metadata changes.

Dataset ID

When processing a collection replacement, ECHO Ingest will fail to replace a record if the new collection attempts to update the dataset ID. This is because the primary identifier of a collection within ECHO, Ingest currently does not support changing the value.

Granule UR

When processing a granule replacement, ECHO Ingest will fail to replace a record if the new granule attempts to update the Granule UR (provider granule ID). This is because the primary identifier of a granule within ECHO, Ingest currently does not support changing the value.

Granule Spatial Representation

When processing a collection replacement, ECHO Ingest will fail to replace a record if the new collection attempts to update the granule spatial representation. ECHO Ingest uses this value to determine which internal geometry tables to store granule spatial information. Switching this spatial representation at the collection level would need to cascade to a reprocessing of all of the granule spatial information. Currently, Ingest does not support changing this value. To make a change, the collection would need to be deleted from ECHO and then re-ingested, along with all of the associated granules. Special case handling can be facilitated by ECHO Operations.

Temporal Range

When processing a collection replacement, ECHO Ingest will fail to replace a record if the new collection's temporal range does include all temporal ranges of its associated granules. Conversely, ECHO Ingest will fail to replace a granule record if the new granule's temporal range is outside of its collection's temporal range.

Collection / Granule Inheritance

When processing a collection or granule replacement, ECHO Ingest will fail to replace a record if one of the following fields is updated such that the required referential inheritance is broken. For instance, if a new additional attribute is added to a granule record that is not found in the collection, the granule replacement will be rejected. Similarly, if an existing additional attribute is removed from a collection record, but a granule exists in that collection with the attribute, the collection replacement will be rejected.

- Campaign
- Platform
- Instrument
- Sensor
- Additional Attributes
- TwoDCoordinateSystem

Partially Adding/Updating/Deleting Items

If you do not want to replace a complete collection or granule record in ECHO, you may specify individual fields to add, update, or remove content. For simplicity, the partial add, update, and delete capability is commonly referred to simply as partial updates and will be hereafter. The partial update format requires full metadata only for the particular field that should be updated along with the ID of the collection or granule record that should be updated. When processing partial updates, ECHO Ingest applies the same principles and validation as when processing inserts. The following sections outline the metadata fields that can be partially added, updated, or deleted.

While performing partial adds, updates or deletes, Ingest will affect different results depending on the nature of the item. In the following sections, each metadata field will have one of the following distinctions identifying how Ingest will perform the partial add, update, or delete.

- **Atomic** - This type of item exists once within a metadata record (e.g. Restriction Flag) and is effectively added, replaced, or removed during partial updates.
- **Unique** - This type of item has a uniquely identifying field and more than one unique instance may occur within a metadata record (e.g. Measured Parameter). Each unique instance may be added, updated, or removed during partial updates.
- **Static** - This type of item may have more than one instance within a metadata record (e.g. Browse Associations), but cannot be updated. These items can only be added or removed during partial updates.

Collections

An ECHO Collection record may have the following elements partially added or updated:

- Visibility (Atomic)
- Temporal (Atomic)
- Spatial (Atomic)
- Restriction Flag (Atomic)
- Delete Time (Atomic)
- Browse image associations (Static)

An ECHO Collection record may have the following elements partially deleted:

- Temporal (Atomic)
- Delete Time (Atomic)
- Restriction Flag (Atomic)
- Spatial (Atomic)
- Specific or All Browse image associations (Static)

Granules

An ECHO Granule record may have the following elements partially added or updated:

1. Visibility (Atomic)
2. Temporal (Atomic)
3. Spatial (Atomic)
4. Day/Night Flag (Atomic)
5. Cloud Cover (Atomic)
6. Delete Time (Atomic)
7. Restriction Flag (Atomic)
8. Online access URL (Unique)
9. Online resource URL (Unique*)
10. Measured Parameter (Unique)
11. Additional Attribute (Unique)
12. Browse image associations (Static)

*Online Resource Urls currently support duplicate entries with the same URL.

An ECHO Granule record may have the following elements partially deleted:

1. Temporal (Atomic)
2. Day/Night Flag (Atomic)
3. Delete Time (Atomic)
4. Restriction Flag (Atomic)
5. Cloud Cover (Atomic)
6. Specific **or** All Online access URL(s) (Unique)
7. Specific **or** All Online resource URL(s) (Unique)
8. Specific **or** All Additional attribute(s) (Unique)
9. Measured Parameter (Unique)
10. Specific **or** All Browse image association(s) (Static)

Browse

Partial adds/updates are not supported for browse

Deleting Items

ECHO Ingest will facilitate the deletion of collection, granule, or browse records from within its holdings. ECHO uses only the item's identification to process the deletion of the item and all the metadata associated with this item. ECHO will keep the deleted items' identification and deletion date in the database for metadata history auditing purposes. The only way to re-install the items in the ECHO system is to re-submit the metadata for those items for insertion.

When a collection is deleted, all associated granules will automatically be deleted. When a browse file is deleted, all collection or granule associations to that browse file will be deleted while the referencing item(s) themselves will remain in ECHO. When collections or granules with browse associations are deleted the browse images will remain in ECHO until an explicit browse image delete has been submitted through ingest. If a browse image being deleted includes a file hosted by ECHO, the image file will be deleted, and is not recoverable.

Code Listing 40. Collection Deletes

```
<CollectionMetaDataFile>
    <CollectionDeletes>
        <CollectionDelete>
            <DataSetId>Insert Additional Attributes Not Unique
V001</DataSetId>
        </CollectionDelete>
    </CollectionDeletes>
</CollectionMetaDataFile>
```

Code Listing 41. Granule Deletes

```
<GranuleMetaDataFile>
    <GranuleDeletes>
        <GranuleDelete>
            <GranuleUR>SC:MOD021KM.004:19250276</GranuleUR>
        </GranuleDelete>
    </GranuleDeletes>
</GranuleMetaDataFile>
```

Code Listing 42. Browse Deletes

```
<BrowseMetaDataFile>
    <BrowseImageDeletes>
        <BrowseImageDelete>
            <ProviderBrowseId>:BR:Browse.001:19250276</ProviderBrowseId>
        </BrowseImageDelete>
    </BrowseImageDeletes>
</BrowseMetaDataFile>
```

Ingest Reaper

The ECHO Collection, Granule, and Browse metadata schema allow for data partners to update their records with a deletion time metadata element. If the value of this field has been set, ECHO Ingest will automatically detect when the time has expired, and will facilitate deletion of the associated record. This automatic deletion will cascade in the same manner as described above. In order to facilitate the automated deletion, ECHO Ingest is configured with an internal reaper which runs on a scheduled basis to clean out items which have passed their deletion time. The frequency which this reaper runs is configured by ECHO Operations. There is no additional notification sent to providers when an item is removed by the reaper. Where possible, Data Partners are encouraged to export explicit deletions instead of setting the deletion time.

Audit History

ECHO Ingest generates an internal audit history recording a subset of information for all metadata items that are deleted during the past 60 days. This history is not available to Data Partners, so any questions regarding item deletions should be coordinated with the ECHO Operations team. The following table shows the metadata fields which are recorded for each type of deletion. Note that the provider time fields are optional because those elements are not a required value in the ECHO metadata model.

Name	Collection	Granule	Browse
Short Name	X	X	
Version ID	X	X	
Long Name	X		
Dataset ID	X	X	
Granule UR		X	
Browse ID			X
Provider Insert Time*	X	X	X
Provider Update Time*	X	X	X
Provider Delete Time*	X	X	X
ECHO Insert Time	X	X	X
ECHO Update Time	X	X	X
ECHO Delete Time	X	X	X

Reconciliation

Through the ECHO 10.0 Ingest Schema, Data Partners may perform two types of metadata reconciliation, *metadata verification* and *inventory verification*. These two verification processes are described below:

- **MetadataVerification** - A full reconciliation of a collection or granule metadata item to include verification of all fields. ECHO will automatically attempt to correct differences within its holdings.
- **Inventory Verification** - A shortened reconciliation mechanism which will allow for the identification of inventory items which are missing from the ECHO holdings or should no longer be held by ECHO. Verified inventory items include collections, granules within a specified collection, or browse records associated with granules within a specified collection.

Both of these methods allow Data Partners to take advantage of Ingest's parallelized data processing and detailed reporting mechanism. The following sections provide a detailed overview of each reconciliation capability.

Metadata Verification

The *Metadata Verification* capability allows an ECHO data partner to perform a full validation on all collection and granule metadata records and fields within the ECHO holdings. For each verified item, ECHO will ensure that every metadata field is correct and will report any inconsistencies discovered. Full verification on browse metadata items is not currently supported due to the initial requirements for this functionality. However, if providers have an interest, this is something that ECHO can consider this for a future release.

In order to utilize this capability, providers must re-export the full collection or granule metadata record, as is generated during normal exports, but with a slightly modified XML structure specifying that the received record should be processed as a verification action. Sample XML blocks for collections and granules are shown below with the new *CollectionVerifications* and *GranuleVerifications* elements. The respective *Collection* and *Granule* elements are repeated as necessary.

Code Listing 43. Collection Metadata Verification Sample

```
<CollectionMetaDataFile>
    <CollectionVerifications>
        <Collection>
            ...
        </Collection>
            ...
    </CollectionVerifications>
</CollectionMetaDataFile>
```

Code Listing 44. Granule Metadata Verification Sample

```
<GranuleMetaDataFile>
    <GranuleVerifications>
        <Granule>
            ...
        </Granule>
            ...
    </GranuleVerifications>
</GranuleMetaDataFile>
```

While verifying a metadata record, ECHO will first determine whether the item exists within ECHO. If it does not, ECHO will treat the item as a metadata insert and attempt to insert the item into its holdings. If the record does exist within ECHO's holdings, then ECHO will perform a detailed comparison of every metadata field and if discrepancies are found ECHO will treat the item as a full metadata replacement and attempt to replace the item in its holdings with the new record. The results of the verification will then be included in the XML ingest report outlining all issues discovered.

This capability should not be used as a means to identify collections or granules which ECHO does not have within its holdings. The *Inventory Verification* capability referenced in the following major section should be used for that purpose. Exports using this *Metadata Verification* can be any subset of a providers holdings.

> Date value comparisons will initially be carried out to the milliseconds without any leniency. It may be possible to expand date comparison leniency if needed.

Discrepancies discovered during metadata verification will be reported as item errors within the Ingest Report and notification email. These item errors will use the following error codes which have been added for this new capability.

- **METADATA_MISMATCH** - Reported if a metadata field did not match between the ECHO holdings and provider's verification granule & collection.
- **COLLECTION_MISSING** - Reported if a collection in the verification listing did not exist in ECHO.
- **GRANULE_MISSING** - Reported if a granule in the verification listing did not exist in ECHO.

When ingest attempts to insert or replace a metadata record which was found to be missing or invalid during verification, it is possible that the subsequent action will fail due to invalid metadata in the verification record. The errors that would be reported include all those that are currently reported during normal ingest inserts and updates. ECHO providers should be sure to analyze the results of a verification export to identify why items have failed. Sample ingest report error messages are shown below for the three new error codes.

Ingest Item Errors

The following item error will be included in the ingest report if a 'field-level' mismatch is discovered.

Code Listing 45. METADATA_MISMATCH field-level error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
        <Message>
            EchoGranule.DeleteTime mismatch
            Expected: 2009-01-05T11:53:50.550Z
            Actual : 2010-01-05T11:53:50.550Z
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an 'object-level' mismatch is discovered. An XML representation of the object is included in the message. An error message has a max length of 1024, so it is possible that the message will be truncated.

Code Listing 46. METADATA_MISMATCH object-level error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
        <Message>
            EchoGranule.AdditionalAttributes mismatch
            Expected: <AdditionalAttributeRef><Name>Name</Name>

<Values><Value>1</Value></Values></AdditionalAttributeRef>
            Actual : [null]
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a 'field-level' mismatch for an object in a list is discovered. In this case, the object which has a mismatching field has a unique identifier (additional attributes, online access URLs, etc.).

Code Listing 47. METADATA_MISMATCH list field-level error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
        <Message>
            EchoGranule.OnlineAccessUrls mismatch
OnlineAccesURL.Description mismatch for: http://provider_url
            Expected: Description Text
            Actual : Incorrect Value
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item errors will be included in the ingest report if a mismatch for objects in a list is discovered. In this case, the objects being compared do not have unique identifiers (online resources, points, etc.) and may produce two messages. One message will indicate that an item was added to one list and another message indicating that an item was deleted from another.

Code Listing 48. METADATA_MISMATCH list object-level item missing error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
        <Message>
            EchoGranule.OnlineResources mismatch
            Expected:
<OnlineResource><URL>missing_url</URL><Description>123</Description></On
lineResource>
            Actual  : [null]
        </Message>
    </ItemError>
</ItemErrorGroup>
```

Code Listing 49. METADATA_MISMATCH list object-level extra item error message

```
<ItemErrorGroup errorCode="METADATA_MISMATCH">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
        <Message>
            EchoGranule.OnlineResources mismatch
            Expected: [null]
            Actual  :
<OnlineResource><URL>extra_url</URL><Description>123</Description></Onli
neResource>
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a matching collection record was not found within ECHO during verification.

Code Listing 50. COLLECTION_MISSING error message

```
<ItemErrorGroup errorCode="COLLECTION_MISSING">
    <ItemError itemType="COLLECTION" itemId="COLLECTION_ID"
level="WARNING">
        <Message>
            Collection was missing from ECHO. An insert attempt will be
made for this collection.
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a matching granule record was not found within ECHO during verification.

CodeListing 51. GRANULE_MISSING error message

```
<ItemErrorGroup errorCode="GRANULE_MISSING">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="WARNING">
        <Message>
            Granule was missing from ECHO. An insert attempt will be
made for this granule.
        </Message>
    </ItemError>
</ItemErrorGroup
```

Ingest Report Processing Totals

The metadata verification activity will be reflected in the processing totals and item errors reported in the Ingest Report and provider notification email. In order to record the verification actions that are performed, the collection and granule processing totals will have an attribute entitled verifications. This attribute will include the number of verification items that were processed, including both successful and unsuccessful verifications. The processingTotals attribute will also include an accounting of the verification actions. Sample processing totals which are included in the Ingest report and notification email are shown below for three possible outcomes.

The following processing totals will be included in an ingest job report and provider notification email for a successful verification of 1000 collection and 1000 granule items.

Code Listing 52. Successful Verification Processing Totals

```
<ProcessingTotals>
    <CollectionProcessingTotals processed="1000" inserted="0"
replaced="0" updated="0" deleted="0" rejected="0" verifications="1000"
inventories="0"/>
    <GranuleProcessingTotals processed="1000" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="1000"
inventories="0"/>
    <BrowseProcessingTotals processed="0" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications'0" inventories="0"/>
</ProcessingTotals>
```

The following processing totals will be included in an ingest job report and provider notification email for a verification of 1000 collection and 1000 granule items where each metadata type had 200 missing items and 200 mismatches. There were no subsequent failures while inserting and replacing these 400 items.

Code Listing 41. Mismatches & Missing Items with No Insertion/Replacement Errors

```
<ProcessingTotals>
    <CollectionProcessingTotals processed="1400" inserted="200"
replaced="200" updated="0" deleted="0" rejected="0" verifications="1000"
inventories="0"/>
    <GranuleProcessingTotals processed="1400" inserted="200"
replaced="200" updated="0" deleted="0" rejected="0" verifications="1000"
inventories="0"/>
    <BrowseProcessingTotals processed="0" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications'0" inventories="0"/>
</ProcessingTotals>
```

The following processing totals will be included in an ingest job report and provider notification email for a verification of 1000 collection and 1000 granule items where each metadata type had 200 missing items and 200 mismatches. There were 100 subsequent failures for each metadata type while inserting the missing 200 items.

Code Listing 42. Mismatches & Missing Items with Insertion/Replacement Errors

```
<ProcessingTotals>
    <CollectionProcessingTotals processed="1400" inserted="100"
replaced="200" updated="0" deleted="0" rejected="100"
verifications="1000" inventories="0"/>
    <GranuleProcessingTotals processed="1400" inserted="100"
replaced="200" updated="0" deleted="0" rejected="100"
verifications="1000" inventories="0"/>
    <BrowseProcessingTotals processed="0" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications'0" inventories="0"/>
</ProcessingTotals>
```

Inventory Verification

The *InventoryVerification* capability allows ECHO data partners to compare a full listing of collection, granule, or browse metadata items between ECHO and their holdings. For each metadata item type, ECHO will perform a two-way comparison for the listing of received items against its holdings. Items which are included in the verification package, but missing in ECHO, will be reported along with items which are in the ECHO holdings, but missing from the verification package. Inventory verification of granules should include a listing of all granules within a single collection. Inventory verification of browse should include a listing of all browse files associated with granules in a specific collection.

In order to utilize this capability, providers must export the full collection, granule, or browse listings in the following XML structure which includes the required elements for uniquely identifying a metadata record. Sample XML blocks for collection, granule, and browse inventory verification are shown below with the new *CollectionInventory*, *GranuleInventories*, and *BrowseInventories* elements. The respective *CollectionRef*, *GranuleInventory*, and *BrowseInventory* elements are repeated as necessary.

Code Listing 42. Collection Inventory Verification XML Sample

```
<CollectionMetaDataFile>
    <CollectionInventory>
        <CollectionReferences>
            <CollectionRef>
                <ShortName></ShortName>
                <VersionId></VersionId>
            </CollectionRef>
            <CollectionRef>
                <DataSetId></DataSetId>
            </CollectionRef>
            ...
        </CollectionReferences>
    </CollectionInventory>
</CollectionMetaDataFile>
```

Code Listing 42. Granule Inventory Verification XML Sample

```
<GranuleMetaDataFile>
    <GranuleInventories>
        <GranuleInventory>
            <CollectionRef>
                ...
            </CollectionRef>
            <GranuleURs>
                <GranuleUR>GRANULE_UR</GranuleUR>
                ...
            </GranuleURs>
        </GranuleInventory>
        ...
    </GranuleInventories>
</GranuleMetaDataFile>
```

Code Listing 42. Browse Inventory Verification XML Sample

```
<BrowseMetaDataFile>
    <BrowseInventories>
        <BrowseInventory>
            <CollectionRef>
                ...
            </CollectionRef>
            <ProviderBrowseIds>
                <ProviderBrowseId>BROWSE_ID</ProviderBrowseId>
                ...
            </ProviderBrowseIds>
        </BrowseInventory>
        ...
    </BrowseInventories>
</BrowseMetaDataFile>
```

As described previously, while verifying a collection, granule, or browse inventory ECHO will identify discrepancies including items which are missing from or should not exist in the ECHO holdings. No corrective actions will be taken as a part of the verification. Discrepancies discovered during inventory verification will be reported as item errors within the Ingest Report and notification email. These item errors will use the following error codes which have been added for this new capability.

- **COLLECTION_MISSING** - Reported if a collection in the verification listing did not exist in ECHO.
- **COLLECTION_UNEXPECTED** - Reported if ECHO has a collection which is not in the verification listing.
- **GRANULE_MISSING** - Reported if a granule in the verification listing did not exist in ECHO within the specified collection inventory.
- **GRANULE_UNEXPECTED** - Reported if ECHO has a granule within the specified collection which is not in the verification listing.
- **BROWSE_LINK_MISSING** - Reported if a browse record in the verification listing did not exist in ECHO associated to the specified inventory collection or granules within that collection.
- **BROWSE_LINK_UNEXPECTED** - Reported if ECHO has a browse record associated with a granule in the specified collection, or the collection itself, which is not in the verification listing.

Sample ingest report error messages are shown below for the new error codes.

Ingest Item Errors

The following item error will be included in the ingest report if a matching collection record was not found within ECHO during inventory comparison.

Code Listing 42. COLLECTION_MISSING Error Message Text

```
<ItemErrorGroup errorCode="COLLECTION_MISSING">
    <ItemError itemType="COLLECTION" itemId="COLLECTION_ID"
level="CRITICAL">
        <Message>
            Inventory mismatch: [Collection] V[1] was not found in ECHO.

        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an extra collection record was found within ECHO during inventory comparison.

Code Listing 43. COLLECTION_UNEXPECTED Error Message Text

```
<ItemErrorGroup errorCode="COLLECTION_UNEXPECTED">
    <ItemError itemType="COLLECTION" itemId="COLLECTION_ID"
level="CRITICAL">
        <Message>
            Inventory mismatch: [Collection] V[1] exists in ECHO, but
was not in the supplied inventory.
        </Message>
    </ItemError>
</ItemErrorGroup>
GRANULE_MISSING Error Message Text
```

The following item error will be included in the ingest report if a matching granule record was not found within ECHO during inventory comparison.

Code Listing 43. GRANULE_MISSING Error Message Text

```
<ItemErrorGroup errorCode="GRANULE_MISSING">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="CRITICAL">
        <Message>
            Inventory mismatch: [Granule UR] does not exist in
collection [Collection] V[1].
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an extra granule record was found within ECHO during inventory comparison.

Code Listing 44. GRANULE_UNEXPECTED Error Message Text

```
<ItemErrorGroup errorCode="GRANULE_UNEXPECTED">
    <ItemError itemType="GRANULE" itemId="GRANULE_UR" level="CRITICAL">
        <Message>
            Inventory mismatch: [Granule UR] exists in [Collection]
V[1], but was not in the supplied inventory.
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if a matching browse record was not found within ECHO during inventory comparison.

Code Listing 45. BROWSE_LINK_MISSING Error Message Text

```
<ItemErrorGroup errorCode="BROWSE_LINK_MISSING">
    <ItemError itemType="BROWSE" itemId="BROWSE_ID" level="CRITICAL">
        <Message>
            Inventory mismatch: [Browse ID] is not currently linked to
collection [Collection] V[1] or any of the collection's granules.
        </Message>
    </ItemError>
</ItemErrorGroup>
```

The following item error will be included in the ingest report if an extra browse record was found within ECHO during inventory comparison.

Code Listing 46. BROWSE_LINK_UNEXPECTED Error Message Text

```
<ItemErrorGroup errorCode="BROWSE_LINK_UNEXPECTED">
    <ItemError itemType="BROWSE" itemId="BROWSE_ID" level="CRITICAL">
        <Message>
            Inventory mismatch: [Browse ID] is currently linked to
collection [Collection] V[1] or any of the collection's granules, but
was not in the supplied inventory.
        </Message>
    </ItemError>
</ItemErrorGroup>
```

Ingest Report Processing Totals

The *inventoryverification* activity will be reflected in the processing totals and item errors reported in the Ingest Report and provider notification email. In order to record the number of inventories reconciled, the collection, granule, and browse processing totals will have a new attribute entitled inventories. This attribute will include the number of complete inventories items that were processed. This does not include the number of complete inventory items which were compared. For instance, if an ingest file contains 100,000 granules for a single collection, the inventories attribute will have a value of 1, not 100,000. ECHO providers should be sure to analyze the results of an inventory reconciliation to identify why items have failed. Sample processing totals which are included in the Ingest report and notification email are shown below for two possible outcomes.

The following processing totals will be included in an ingest job report and provider notification email for a successful verification of a provider's 1000 collections, 3 granule inventories, and 3 browse inventories. Note that the collection processing totals' value for the new *inven tories* attribute will always be 1 since there is only one inventory of collections.

Code Listing 47. Successful Verification Processing Totals

```
<ProcessingTotals>
    <CollectionProcessingTotals processed="1" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="1"/>
    <GranuleProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="3"/>

    <BrowseProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="3"/>
</ProcessingTotals>
```

The following processing totals will be included in an ingest job report and provider notification email for a verification of a provider's 1000 collections, 3 granule inventories, and 3 browse inventories, where 1 of each of the granule and browse inventories referenced invalid

collections.

Code Listing 48. Inventory Verification with Rejections

```
<ProcessingTotals>
    <CollectionProcessingTotals processed="1" inserted="0" replaced="0"
updated="0" deleted="0" rejected="0" verifications="0" inventories="1"/>
    <GranuleProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="1" verifications="0" inventories="2"/>
    <BrowseProcessingTotals processed="3" inserted="0" replaced="0"
updated="0" deleted="0" rejected="1" verifications="0" inventories="2"/>
</ProcessingTotals>
```

Ingest Errors

During Ingest processing, the following errors may be encountered:

- **JobErrors** - Created when ECHO cannot process the job in its entirety.
- **FileErrors-** Created when ECHO cannot parse or validate the input file.
- **ItemErrors-** Created when ECHO can parse the input file but finds the data invalid.

Any error that is generated will have a level associated with it: CRITICAL or WARNING. In general, critical errors indicate that the job or part of the job was rejected by ECHO Ingest. Errors that are warnings indicate that some abnormality was detected but the job or affected part of the job may have been ingested dependent on the absence of other critical errors. Each of these errors is discussed in more detail below. For more information regarding the Ingest Summary Report which is referenced, please refer to section 5.8.1 of this document. For a full list of possible errors review the online Ingest documentation at: http://www.echo.nasa.gov/ingest/schemas/operations/docs/.

Job Errors

Initial scanning of a job by ECHO Ingest may reveal problems that will stop processing for that job immediately. The Ingest Summary Report will contain the necessary information to determine why the job was canceled.

DUPLICATE_SEQUENCE_NUMBER Error

If a processed package's sequence number is less than the current sequence number being processed a DUPLICATE_SEQUENCE_NUMBER job error will be generated.

Code Listing53. Job Error Message

```
<JobErrors>
<Error errorCode="DUPLICATE_SEQUENCE_NUMBER" level="CRITICAL"> <Message>
The sequence number [300258] is less than or equal to the last sequence
number seen for the provider.
</Message> </Error>
</JobErrors>
```

MANIFEST_MISSING Error

If an Ingest package does not include the manifest.xml file either due to non-existance or an incorrect naming, a MANIFEST_MISSING job error will be generated.

Code Listing 54. Missing Manifest File Error

```
<JobErrors>
    <Error errorCode="MANIFEST_MISSING" level="CRITICAL">
        <Message>
            No [manifest.xml] file found in package
        </Message>
    </Error>
</JobErrors>
```

File Errors

Before ECHO ingests your metadata, it analyzes and validates the metadata against the ECHO Format. If an error is detected, ECHO may create an input file error or an item error.

FULL_SCHEMA Error

If an input file does not pass validation against the ECHO Format a FULL_SCHEMA file error will be generated. To allow the ingest of files that may have some valid and some invalid items, FULL_SCHEMA errors are considered warnings. An input file with a FULL_SCHEMA error will be subjected to validation of each individual item against the ECHO Format to determine if any items within the input file are valid.

Code Listing 55. FULL_SCHEMA Error Message

```
<FileErrors>
    <Error errorCode="FULL_SCHEMA" level="WARNING">
        <Message>
            Line:180 Col:140, cvc-minLength-valid: Value '' with length
= '0' is not facet-valid with respect to minLength '1' for type
'GranuleUR'.
        </Message>
    </Error>
</FileErrors>
```

The error message indicates the problem line and column number in the input file. The precise message will vary depending on the type of error found. The sample above indicates that a zero-length GranuleUR was submitted and the ECHO Format specifies that GranuleUR has a minimum length of 1.

STRUCTURAL_SCHEMA Error

If an input file is marked with a FULL_SCHEMA error ECHO Ingest will attempt to verify that the input file can be parsed by validating the input file conforms the ECHO Format structurally. If an input file is not structurally valid a STRUCTURAL_SCHEMA error will be generated. STRUCTUAL_SCHEMA errors are critical. If an input file is structurally valid then each metadata item will be subject to validation against the full ECHO Format, in order to ingest any valid metadata items.

Code Listing 56. STRUCTURAL_SCHEMA Error Message

```
<FileErrors>
    <Error errorCode="FULL_SCHEMA" level="WARNING">
        <Message>
            Line:66 Col:22, cvc-complex-type.2.3: Element
'GranulePartialAdds' cannot have character [children], because the
type's content type is element-only.
        </Message>
    </Error>
    <Error errorCode="STRUCTURAL_SCHEMA" level="CRITICAL">
        <Message>
            Line:66 Col:22, cvc-complex-type.2.3: Element
'GranulePartialAdds' cannot have character [children], because the
type's content type is element-only.
        </Message>
    </Error>
</FileErrors>
```

In the sample above the input was structurally invalid with a character data present in the GranulePartialAdds tag. Such a file cannot be parsed by ECHO Ingest and so a STRUCTURAL_SCHEMA error is generated and the entire input file is rejected.

SCHEMA_VALIDATION_ERROR Error

If an input file was not valid against the full ECHO Format but was structurally valid then each item will be individually validated against the full ECHO Format. A SCHEMA_VALIDATION_ERROR will be generated for any item failing validation and the item will not be ingested. Otherwise, items passing validation will continue normal processing by ECHO Ingest.

Code Listing 57. SCHEMA_VALIADATION_ERROR Error Message

```
<ItemErrorGroups>
    <ItemErrorGroup errorCode="SCHEMA_VALIDATION_ERROR">
        <ItemError itemType="COLLECTION" itemId="MLS/Aura L2
Diagnostics, Miscellaneous Grid V001" level="CRITICAL">
            <Message>
                Line:14 Col:141, cvc-minLength-valid: Value '' with
length = '0' is not facet-valid with respect to minLength '1' for type
'ShortName'.
            </Message>
         </ItemError>
    </ItemErrorGroup>
</ItemErrorGroups>
```

In the sample above the offending collection could be identified and the dataset ID was included in itemId. The collection was not ingested because the ShortName supplied was an empty string and the ECHO Format requires a minimum ShortName length of 1.

FILE_TYPE_INDETERMINABLE Error

If an input file is invalid XML, a FILE_TYPE_INDETERMINABLE error will be generated. ECHO Ingest will also generate a FILE_TYPE_INDETERMINABLE error if an input file is valid XML but ECHO Ingest cannot determine the metadata item type of the file. FILE_TYPE_INDETERMINEABLE errors are critical.

Code Listing 58. FILE_TYPE_INDETERMINABLE Error Message

```
<FileErrors>
    <Error errorCode="FILE_TYPE_INDETERMINABLE" level="CRITICAL">
        <Message>
            Could not determine file type from input file
[EPGMOLT200726120072620101.20070919004641.xml]
        </Message>
    </Error>
</FileErrors>
```

Item Errors

All items valid against the ECHO Format are also subject to data integrity validation according to ECHO business rules. This includes inserts, replacements, and updates to existing items. For each data integrity error discovered, ECHO Ingest will include the type of error, the invalid item ID, and the type of item. All data integrity errors are critical errors. For a comprehensive list of the errors which may occur during Ingest, refer to the ECHO Ingest Summary Report specification available on the ECHO website (http://www.echo.nasa.gov/ingest/schemas/operations/docs/).

All data integrity errors will identify the invalid item using itemId and include a detailed message regarding the cause of the error. The precise message will vary depending on the type of error. An overview of some common item errors is given in the following section.

GRANULE_NOT_EXISTS Error

If a granule deletion, partial update, or partial deletion references a granule which does not exist in the ECHO holdings, ECHO will return a GRANULE_NOT_EXISTS error and not process the specified record.

Code Listing 59. Referenced Granule Does Not Exist

```
<ItemErrorGroups>
    <ItemErrorGroup errorCode="GRANULE_NOT_EXISTS">
        <ItemError itemType="GRANULE" itemId="SC:g3asspb.004:24162750"
level="CRITICAL">
            <Message>
                Validation error, SC:g3asspb.004:24162750. Granule does
not exist.
            </Message>
        </ItemError>
    </ItemErrorGroup>
</ItemErrorGroups>
```

OUT_OF_DATE Error

If the date of a collection or granule in the ECHO database is more recent than the date on the same collection or granule in your input, ECHO will return an OUT_OF_DATE error and not process the specific record.

Code Listing 60. Input Data Older than the Current Record

```
<ItemErrorGroups>
    <ItemErrorGroup errorCode="OUT_OF_DATE">
        <ItemError itemType="COLLECTION" itemId="Collection1 V001"
level="CRITICAL">
            <Message>
                Validation error, new last update date Tue Oct 10
13:00:00 EDT 2000 is before current last update date Thu Oct 10 13:00:00
EDT 2002
            </Message>
        </ItemError>
    </ItemErrorGroup>
</ItemErrorGroups>
```

BROWSE_NOT_EXISTS Error

If a granule or collection browse linking partial update or browse deletion references a browse record which does not exist in the ECHO holdings, ECHO will return a BROWSE_NOT_EXISTS error and not process the specified record.

Code Listing 61. Referenced Browse Does Not Exist

```
<ItemErrorGroups>
<ItemErrorGroup errorCode="BROWSE_NOT_EXISTS">
<ItemError itemType="BROWSE" itemId=":SC:MISBR.005:6561341:1.HDF-EOS"
level="CRITICAL">
<Message>
Validation error,SC:MISBR.005:6561341:1.HDF-EOS does not exist.
</Message>
</ItemError> </ItemErrorGroup>
</ItemErrorGroups
```

COLLECTION_REF_INVALIDError

If a granule insert references a collection record which does not exist in the ECHO holdings, ECHO will return a COLLECTION_REF_INVALID error and not process the specified record.

Code Listing 62. Collection Referenced by Input Granule(s) Does Not Exist

```
<ItemErrorGroups>
<ItemErrorGroup errorCode="COLLECTION_REF_INVALID">
<ItemError itemType="GRANULE" itemId="ScInputGranule001"
level="CRITICAL"> <Message>
Referenced collection by short name [Non-Existant-Collection] and
version [1] does not exist
</Message> </ItemError>
</ItemErrorGroup> </ItemErrorGroups>
```

SPATIAL_INVALID Error

If a granule or collection insert or update contain invalid spatial data, ECHO will will return a SPATIAL_INVALID error and not process the specified record. For information on spatial rules for avoiding invalid spatial errors, refer to section 3 of this document.

Code Listing 63. Invalid Spatial Coverage Area Data

```
<ItemErrorGroups>
    <ItemErrorGroup errorCode="SPATIAL_INVALID">
        <ItemError itemType="COLLECTION" itemId="MISR Level 2 Aerosol
parameters V002" level="CRITICAL">
            <Message>
                Validation error, Spatial Validation Error
BOUNDING_RECTANGLE #1 [13356 [Element <1>] [Coordinate <1>][Ring <1>]]
            </Message>
        </ItemError>
    </ItemErrorGroup>
</ItemErrorGroups>
```

TEMPORAL_INVALID_DATE_RANGE Error

If a granule insert or partial update changes the granule's temporal range to a value which is outside of its collections temporal range, ECHO will return a TEMPORAL_INVALID_DATE_RANGE error and not process the specified record.

Code Listing 64. Data Integrity Error Message

```
<ItemErrorGroup errorCode="TEMPORAL_INVALID_DATE_RANGE"> <ItemError
itemType="GRANULE"
itemId="sample.granule.dat" level="CRITICAL">
<Message>
The granule's start time [1994-04-09T12:00:00Z] is before the
collection's [1994-08-16T12:00:00Z]
</Message> </ItemError>
</ItemErrorGroup>
```

Ingest Reporting

Ingest Summary Report

At the completion of each Ingest Job, Ingest will automatically generate an Ingest Summary Report in XML format for the job that details a job's activities as well as any abnormalities discovered in the input file(s). This report will still be created in the event that the job is deleted by an Ingest Operator. The reports are designed to be machine-readable and therefore may not be formatted for human consumption. The reports can be used to perform a post processing analysis of a completed job.

The Ingest Summary Report will be available in the designated provider report directory (located as agreed upon with ECHO Operations). This directory will be accessible from the ftp site for you to download reports and will be archived in the job directory for the Ingest Operator. (Data Partners will not have access to the job directory. If access to a report is needed after it has been removed from the ftp site, ECHO Operations should be contacted.) ECHO Operations will periodically cleanup Ingest reports older than 60 days from the FTP site.

For a full description of the Ingest Detail Report, refer to the schema and documentation referenced on the ECHO Website (http://www.echo.nasa.gov/ingest/schemas/operations/docs/index.html).

The Ingest Detail Report is applicable to the following types of items:

- Collection Inserts, Updates, Replacements, Deletions, and Reconciliation

- Granule Inserts, Updates, Replacements, Deletions, and Reconciliation
- Browse Inserts, Replacements, Deletions, and Reconciliation

The Ingest Detail Report consists of two parts: overview and details. Both of these sections are described below. For an example of a full Ingest Summary Report, see Appendix H.

Ingest Detail Report Overview

The overview section of the Ingest Detail Report provides a high level summary of what actions were performed while processing the ingest job.
includes the following information:

- Job Start & End Date
- Job Name (If Provided)
- Sequence Number (If Provided)
- Collection, Granule, and Browse Processing Totals including:

    a. Total Processed - The number of ingest actions performed including item inserts, updates, and deletions, collection/granule/browse inventory existence checks(+1 per inventory), collection/granule metadata verification (+1 per item).
    b. Total Inserted - The number of items inserted. This will include explicit inserts processed and items that are inserted as a part of the correct metadata verification actions.
    c. Total Replaced - The number of items replaced. This will include explicit replacements processed and items that are successfully replaced as a part of the correct metadata verification actions.
    d. Total Updated - The number of items that were partially updated, including partial adds and partial deletes.
    e. Total Deleted - The number of items which were deleted from the ECHO inventory. This number will not reflect items which have had their <DeleteTime> metadata value set for future deletion by the ECHO Ingest reaper.
    f. Total Rejected - The number of items for which an insert, replacement, update or deletion was rejected. When performing inventory verification, each inventory that cannot be processed will be listed as a single rejection. When performing metadata verification, each item which cannot be verified, or which fails a subsequent corrective insert or replacement action.
    g. Total Inventories - The number of inventories which verification was successfully performed, irrespective of how many items were found to be missing or unexpected. Since there is only 1 collection inventory, this field will report a '1' if collection inventory verification is performed. For granules and browse, this field will contain the number of inventories that were compared. This will **not** include the number of items in each inventory that were compared.
    h. Total Verifications - The number of items for which metadata verification was successfully performed, irrespective of whether the items were found to be matching or not.

- Job-level errors encountered during Ingest

Code Listing 65. Example of the Ingest Summary Report Overview Section

```
<Overview sequenceNumber="5734" endDate="2009-05-06T08:11:57.535-04:00"
startDate="2009-05-06T08:07:51.703-04:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals processed="0" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0" inventories="0"
verifications="0"/>
        <GranuleProcessingTotals processed="604" rejected="2"
deleted="0" updated="240" replaced="0" inserted="362" inventories="0"
verifications="0"/>
        <BrowseProcessingTotals processed="240" rejected="0" deleted="0"
updated="0" replaced="0" inserted="240" inventories="0"
verifications="0"/
    </ProcessingTotals>
</Overview>
```

Ingest Detail Report Details

The following details for each file will be included:

1. File Name (this is the original file name as sent by the Data Provider)
2. Collection, Granule, and Browse Processing Totals including:
    a. Total Processed - The number of ingest actions performed including item inserts, updates, and deletions, collection/granule/browse inventory existence checks(+1 per inventory), collection/granule metadata verification (+1 per item).
    b. Total Inserted - The number of items inserted. This will include explicit inserts processed and items that are inserted as a part of the correct metadata verification actions.
    c. Total Replaced - The number of items replaced. This will include explicit replacements processed and items that are successfully replaced as a part of the correct metadata verification actions.
    d. Total Updated - The number of items that were partially updated, including partial adds and partial deletes.
    e. Total Deleted - The number of items which were deleted from the ECHO inventory. This number will not reflect items which have had their <DeleteTime> metadata value set for future deletion by the ECHO Ingest reaper.
    f. Total Rejected - The number of items for which an insert, replacement, update or deletion was rejected. When performing inventory verification, each inventory that cannot be processed will be listed as a single rejection. When performing metadata verification, each item which cannot be verified, or which fails a subsequent corrective insert or replacement action.
    g. Total Inventories - The number of inventories which verification was successfully performed, irrespective of how many items were found to be missing or unexpected. Since there is only 1 collection inventory, this field will report a '1' if collection inventory verification is performed. For granules and browse, this field will contain the number of inventories that were compared. This will **not** include the number of items in each inventory that were compared.
    h. Total Verifications - The number of items for which metadata verification was successfully performed, irrespective of whether the items were found to be matching or not.
3. File errors encountered during Ingest
4. Item errors encountered during Ingest including:
    a. Error Code
    b. List of items associated with each error code including:
        i. Item ID
            1. Dataset ID for collections
            2. Granule UR for granules
            3. Browse ID for Browse
        ii. Item Type
        iii. Error Level
5. Rejection Reason

Code Listing 66. Example of the Ingest Summary Report Details Section

```
<Details>
    <MetadataFiles>
        <MetadataFile
name="EchoDel.acdisc_main_ops.T20090506101026_0.xml">
            <ProcessingTotals>
                <CollectionProcessingTotals processed="0" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0" />
                <GranuleProcessingTotals processed="2" rejected="2"
deleted="0" updated="0" replaced="0" inserted="0" />
                <BrowseProcessingTotals processed="0" rejected="0"
deleted="0" updated="0" replaced="0" inserted="0" />
            </ProcessingTotals>
            <FileErrors />
            <ItemErrorGroups>
                <ItemErrorGroup errorCode="GRANULE_NOT_EXISTS">
                    <ItemError itemType="GRANULE" itemId="
SOR3TSI6.009:sorce_tsi" level="CRITICAL">
                        <Message>
                            Validation error, SOR3TSI6.009:sorce_tsi
granule does not exist.
                        </Message>
                    </ItemError>
                </ItemErrorGroup>
            </ItemErrorGroups>
        </MetadataFile>
    </MetadataFiles>
</Details>
```

IngestEmails

ECHO Ingest is configured by ECHO Operations to notify one or more Data Partner personnel regarding Ingest processing. The following situations will generate an email to the configured addresses.

- **IngestStartup/Shutdown** - When ECHO Operations starts up or shuts down Ingest.
- **JobStart** - When an Ingest job is detected by ECHO Ingest and queued for processing.
- **JobCompletion** - When an Ingest job completes processing.
- **ManualProviderPause/Resume** - When ECHO Operations manually pauses or resumes an Ingest provider.
- **ErrorConditionProviderPause/Resume** - When ECHO Ingest pauses an ingest provider due to an internal error. ECHO Operations is also notified with the error information for analysis.
- **OutofSequence** - When an Ingest job has been queued but connot be processed because a prior sequenced job is expected. ECHO Operations configures this warning threshold in conjunction with each data partner.
- **NoInputReceived** - When no Ingest jobs have been detected for a configurable amount of time. ECHO Operations configures this warning threshold in conjunction with each data partner.

ECHO Ingest Accounting Tool

Data Partners may use the ECHO Ingest Accounting Tool (EIAT) in order to view the current status of submitted Ingest jobs and summary information for completed jobs. The EIAT can be access via the following URLs. Login credentials require a username and password for a user which has been assigned 'provider role' for an ECHO provider in the associated mode.

- **Operations**- http://www.echo.nasa.gov/eiat
- **PartnerTest** - http://eiat-test.echo.nasa.gov

Ingest Configuration

General Ingest Configuration

ECHO Operations will configure the following items for all ingest activity. Each of these values will be coordinated with data partners and adjusted as necessary to support operational Ingest.

- **OutOfSequenceThreshold** - The maximum amount of time that ingest will wait if receiving a job out of sequence for the other jobs in sequence before notifying Ingest Operators and the Provider contacts.
- **ReaperInterval** - The frequency at which the Ingest internal reaper will run to remove Collections, Granules, and Browse which have been marked with an expired deletion time.
- **ResourceWaitTime** - The maximum amount of time that can elapse without receipt of resources for a file (i.e., browse images) prior to Ingest continuing in processing.
- **InputQuietTime** - The amount of time that a file must have no modifications and be complete, before it is added it to the Job. E.g., file FOO.XML received from GSFC must not have been modified for the past 10 minutes before it is added to the Job.

Provider Specific Configuration

ECHO Operations will configure the following items on a per-provider basis. Each of these values will be coordinated with data partners and adjusted as necessary to support operational Ingest.

- **ProviderProcessingThreads(Pipelines)** - The number of processing threads dedicated to a provider's metadata ingest.
- **ProviderEmailAddresses** - Data Partner email addresses that will receive automated Ingest notification emails.
- **ReportDirectory** - The ftp directory into which Ingest Summary Reports will be placed upon job completion.
- **InputDirectory** - The ftp directory from which Ingest will recursively search for metadata files and packages to process.
- **DeliveryMechanism** - The Ingest delivery mechanism (Package or Single-File) which will be detected for job creation.
- **ScanInterval** - The frequency that Ingest will scan the input directory for received Ingest packages or files.
- **NoInputReceivedThreshold** - The maximum amount of time that can elapse without receipt of Ingest input before the Ingest Operator and Provider are notified that no input has been received.